Better PN Generators for CDMA Application – A Verilog-HDL Implementation Approach

Afaq Ahmad

Sultan Qaboos University, Department of Electrical & Computer Engineering, College of Engineering
P. O. Box 33, Postal Code 123Muscat, Oman
afaq@squ.edu.om

***Abstract*-Pseudo Noise (PN) sequence generator is one of the important element in the designing of Code Division Multiple Access (CDMA) system. To spread spectrum CDMA applications each user is assigned with a PN sequence for the purpose of spreading and dispreading. Various PN sequences can be generated using Linear Feedback Shift Register (LFSR). For an n-bit LFSR not all the characteristic polynomials but only a few will be able to provide PN sequences. In practice a proper and predefined characteristic polynomial of PN generators is used in CDMA systems. However, LFSR circuits are implemented with VLSI technology and a PN sequence generator design may vary in respect to power dissipation, area and propagation delay. Thus, in CDMA systems a careful selection of PN generators is essential. This paper presents a search procedure to obtain a list of characteristic polynomials so that n-bit LFSRs can be implemented with minimum hardware area. In this paper we also discuss the implementation of such PN generators using Verilog HDL.**

***Keywords*-Pseudo Noise (PN) Sequence; LFS; CDMA; Verilog-HDL; Shift Register Feedback Taps; Power Dissipation; Pseudo-Random Binary Sequences**

I. INTRODUCTION

Code Division Multiple Access (CDMA) is a spread spectrum multiple access technique. A radical and revolutionary new concept in wireless communications is none but known as CDMA. The basic idea of CDMA is to use a wide frequency band common to all the users, whose signals are made distinguishable by assigning mutually uncorrelated code modulation patterns to the various users. Communication systems for point-to-point military communications using this basic type of code modulation have been put into use since the 1960's. They were known as Direct Sequence Spread Spectrum (DSSS) systems. But CDMA was invented to handle the multi-user situation with many telephones transmitting simultaneously. Using CDMA technique by cellular radio system operators as an upgrade dramatically increases both their system capacity and the service quality, thereby, CDMA technique has gained widespread international acceptance. CDMA has a history that goes back to the early days of World War II. The techniques have been used in military applications for many years. Since CDMA is a form of spreading spectrum, thus, the principle is based on the use of noise-like carrier waves, and as the name implies, bandwidths are much wider than that which requires simple point-to-point communication at the same data rate. At the beginning of the technique there were two motivations, (a) either to resist enemies' efforts to jam the communication (anti-jam), or (b) or to hide the fact that communication is actually taking place, which is sometimes called Low Probability of Intercept (LPI) [1] – [6].

With the advancement of VLSI technology, spread spectrum CDMA system has now risen up as a highly emerging digital technology for 3G and beyond mobile systems. In this system the basic hardware involves maximal length of PN sequence or m-sequence generated by a Linear Feedback Shift Register (LFSR). A LFSR is basically a shift register configuration that propagates the stored patterns. A LFSR can generate m-sequences provided that the characteristic polynomial of this LFSR is represented by a primitive polynomial. Thus it can be summarized that PN sequence generation is considered to be the heart of CDMA system. The maximal length PN sequence (m-sequence) is described as the best-known PN sequence whose length is equal to its period (p). Various PN codes can be generated using different structures of LFSR corresponding to different characteristic polynomials. An m-sequence of period 'p' can be realized by an n-bit PN sequence generators equipped with different characteristic polynomial. Each characteristic polynomial has different functional relation with feedback taps for the LFSR circuit. Therefore, the role of feedback taps in implementation of the LFSR circuits with VLSI technology makes it a very important and useful candidate in the design of low-power communication systems [1] – [9].

The current generation's wireless CDMA technology uses 'Gold' codes as the scrambling code. Gold codes are obtained by combining two PN sequences with modulo-2 adding, or performing XOR operation on two of the outputs together. These codes have specific cross-correlation properties to allow as many users as possible with minimum interference [1] – [9].

Pseudo Noise sequences are generated with multiple sets of shift register feedback taps amounting to construct such a PN generator of the same order. For example, a 6-bit linear feedback shift having registered feedback taps 1, 6 and 1, 3, 4, 6 described by the primitive polynomials $x^6 + x^3 + 1$ and $x^6 + x^4 + x^3 + x + 1$ are capable of generating m-sequences. Both structures of PN generators require different hardware areas. Hence judicious selection of set of shift register feedback taps can minimize hardware area. This is the prime idea of carrying out this research.

II. LINEAR FEEDBACK SHIFT REGISTER

In general, by using two or more D-type flip-flops and one or more exclusive-OR (XOR) gate, an LFSR can be realized. An n-bit LFSR is depicted in Fig. 1. The flip-flops form a linear shift register whereas XOR gates are responsible for shift register feedbacks. Whether an LFSR shifts right or left does not really matter and is usually determined by the requirements of the circuit that the LFSR is driving or the method that it is being constructed by using either internal XOR or external XOR arrangements. In applications, the shift register can be preset to a known initial condition. But in general they are either set to all zeros except for one bit, or set

to all ones except for one bit. If XOR circuits are used to generate the feedback input to the shift register, then the state of all zeros is not allowed as the system would never leave the all zero state.

Linear Feedback Shift Registers are not truly random generators because after a certain number of cycles, the cycle will repeat, hence they are also termed "pseudo-random generators". The maximum number of cycles before the cycle repeats can be computed as (2^n)-1 where n represents the number of flip-flops used in the LFSR [10] - [35].

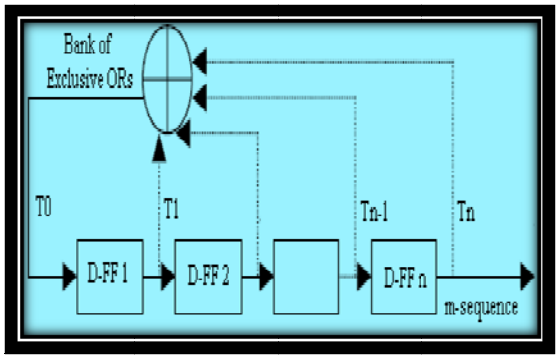

Fig. 1 An n-bit LFSR

A given set of shift register feedback taps can be expressed in a form of polynomial and known as characteristic polynomial of an LFSR structure. Any LFSR structure can be represented as a polynomial of variable x. Thus, characteristic polynomial of an n-bit LFSR T(x) can be given as described in Equation (1).

$$T(x) = T_n x^n + T_{n-1} x^{n-1} + \ldots + T_2 x^2 + T_1 x^1 + T_0 x^0 \quad (1)$$

Where $\{T_i\}$ are the shift register feedback taps with weights either 0 or 1. If particular shifts register's feedback tap exist then 1 otherwise 0. The taps T_n and T_0 are always 1. The necessary condition for generator polynomial T(x) to generate an m-sequence is that T(x) be irreducible. But this condition is not sufficient. The irreducible polynomials which are primitive can only generate m-sequence. So the necessary and sufficient condition for generation of m-sequence is that the generator polynomial should be the primitive characteristic polynomial.

A polynomial T(x) of degree n over GF(2) is irreducible if it cannot be factored into non-trivial polynomials i.e. if T(x) is not divisible by any polynomial. On the other hand an irreducible polynomial T(x) of degree n is said to be primitive if it has order $2^n - 1$ i.e. if the smallest positive integer m for which T(x) divides $x^m + 1$ is $m = 2^m - 1$. An important aspect of irreducible and primitive polynomial is that all the irreducible polynomials are primitive but the reverse is not true. For example, the polynomial $x^4 + x^3 + 1$ is irreducible as well as primitive where as the polynomial $x^4 + x^3 + x^2 + x + 1$ is irreducible but not primitive.

Table 1 illustrates how the patterns are produced by the LFSR which has feedback tapped from 1st and 3rd ($T_0 = T_1 = T_4 = 1$; characteristic polynomial $T(x) = 1 + x + x^4$) also, assuming that the pattern of 111 is used as an initial loading (seed) of LFSR. It can be visualized as the pattern 16 becomes the pattern 1 and repetition starts thus the period of LFSR sequence is 15.

TABLE 1

LFSR SEQUENCE GENERATED T(X) = 1 + X +X4)

Clock	D-FF1	D-FF2	D-FF3	D-FF4
1	1	1	1	1
2	0	1	1	1
3	1	0	1	1
4	0	1	0	1
5	1	0	1	0
6	1	1	0	1
7	0	1	1	0
8	0	0	1	1
9	1	0	0	1
10	0	1	0	0
11	0	0	1	0
12	0	0	0	1
13	1	0	0	0
14	1	1	0	0
15	1	1	1	0
16	1	1	1	1

III. PRIMITIVE POLYNOMIALS WITH MINIMUM, MAXIMUM TAPS

A PN sequence generator of size n may have a set of shift register feedback with maximum or minimum taps. Using the maximum shift register feedback taps leads to the worst design whereas the minimum one is known as one of the best.

There is no quick way to determine if a set of shift register feedback taps in an n-bit LFSR can or cannot generate a sequence of maximal length. However, there are some ways to ease the search. Some of the points and properties of maximal length sequence can be exploited to speed up the search process. For example, a set of an odd number feedback taps entries cannot generate the maximal length sequence. Similarly, it is also a point of notice that the tap values in a maximal length shift register feedback tap sequence are all relatively prime. A shift register feedback tap sequence like 12, 9, 6, and 3 will not be maximal length sequence generator in a 12-bit LFSR because the shift register feedback tap values are all divisible by 3. Considering all such points and properties an algorithm is designed and given below:

Algorithm

STEP 0:

For an n-bit LFSR, generate vector $T = [T_1, T_2, T_3, .., T_i.., T_n] = [1, 1, \ldots, 1, \ldots, 1]$.

STEP 1:

Check, for number of entries in T; if odd go to STEP 2; else, go to STEP 12.

STEP 2:

(a)Initialize periodicity counter PC = 0, and,

(b)Initialize appending index AI = 1.

STEP 3:

In T check, is (AI) = 1? If no go to STEP 4; else, go to STEP 5.

STEP 4:

Check, is AI = n? If no, increment AI, and, go to STEP 3; else, go to STEP 10.

STEP 5:

Modify PC as PC = PC + (AI +1); and check, is PC = 2^n - 1? If yes, go to STEP 11; else, go to STEP 6.

STEP 6:

Generate vector T1 by shifting right T to (AI+1) bits.

STEP 7:

Generate vector T2 by deleting the right (AI+1) bits of T and then, appending (AI+1) zeros to the left side of remaining T.

STEP 8:

Modify T; T = T1⊕ T2.

STEP 9:

Go to STEP 2 (b).

STEP 10:

Check, is PC = 2^n - 1? If yes, go to STEP 11; else, go to STEP 12.

STEP 11:

Declare that the LFSR equipped with shift register feedback tap (T) can generate maximal length sequence and go to STEP 13.

STEP 12:

The LFSR equipped with shift register feedback tap (T) cannot generate maximal length sequence.

STEP 13: Stop.

TABLE 2
PRIMITIVE CHARACTERISTIC POLYNOMIALS (N = 2 TO 6)

n	2-Taps	4-Taps
2	x^2+x+1	
3	x^3+x+1	
	x^3+x^2+1	
4	x^4+x+1	
	x^4+x^3+1	
5	x^5+x^2+1	$x^5+x^3+x^2+x+1$
	x^5+x^3+1	$x^5+x^4+x^2+x+1$
		$x^5+x^4+x^3+x+1$
		$x^5+x^4+x^3+x^2+1$
6	x^6+x+1	$x^6+x^5+x^4+x^3+x+1$
	x^6+x^5+1	$x^6+x^5+x^2+x+1$
		$x^6+x^5+x^3+x^2+1$
		$x^6+x^5+x^4+x+1$

TABLE 3
PRIMITIVE CHARACTERISTIC POLYNOMIALS (N = 7)

n	2-Taps	4-Taps	6-Taps
7	x^7+x+1	$x^7+x^3+x^2+x+1$	$x^7+x^5+x^4+x^3+x^2+x+1$
	x^7+x^3+1	$x^7+x^4+x^3+x^2+1$	$x^7+x^6+x^5+x^3+x^2+x+1$
	x^7+x^4+1	$x^7+x^5+x^2+x+1$	$x^7+x^6+x^5+x^4+x^2+x+1$
	x^7+x^6+1	$x^7+x^5+x^3+x+1$	$x^7+x^6+x^5+x^4+x^3+x^2+1$
		$x^7+x^5+x^4+x^3+1$	
		$x^7+x^6+x^3+x+1$	
		$x^7+x^6+x^4+x+1$	
		$x^7+x^6+x^4+x^2+1$	
		$x^7+x^6+x^5+x^2+1$	
		$x^7+x^6+x^5+x^4+1$	

TABLE 4
PRIMITIVE CHARACTERISTIC POLYNOMIALS (N = 8)

n	4-Taps	6-Taps
8	$x^8+x^4+x^3+x^2+1$,	$X^8+x^6+x^4+x^3+x^2+x+1$
	$x^8+x^5+x^3+x+1$	$X^8+x^7+x^6+x^3+x^2+x+1$
	$x^8+x^5+x^3+x^2+1$,	$X^8+x^7+x^6+x^5+x^2+x+1$
	$x^8+x^6+x^3+x^2+1$	$X^8+x^7+x^6+x^5+x^4+x^2+1$
	$x^8+x^6+x^5+x+1$,	
	$x^8+x^6+x^5+x^2+1$	
	$x^8+x^6+x^5+x^3+1$	
	$x^8+x^6+x^5+x^4+1$,	
	$x^8+x^7+x^5+x^2+x+1$	
	$x^8+x^7+x^3+x^2+1$	
	$x^8+x^7+x^5+x^3+1$	
	$x^8+x^7+x^6+x+1$	

TABLE 5
PRIMITIVE CHARACTERISTIC POLYNOMIALS (N = 9)

n = 9	
2 Taps	x^9+x^4+1, x^9+x^5+1
4 Taps	$x^9+x^4+x^3+x+1$, $x^9+x^5+x^3+x^2+1$, $x^9+x^5+x^4+x+1$, $x^9+x^6+x^4+x^3+1$, $x^9+x^6+x^5+x^3+1$, $x^9+x^7+x^2+x+1$, $x^9+x^7+x^4+x^2+1$, $x^9+x^7+x^5+x+1$, $x^9+x^7+x^5+x^2+1$, $x^9+x^7+x^6+x^4+1$, $x^9+x^8+x^4+x+1$, $x^9+x^8+x^4+x^2+1$, $x^9+x^8+x^5+x+1$, $x^9+x^8+x^5+x^4+1$, $x^9+x^8+x^6+x^5+1$, $x^9+x^8+x^7+x^2+1$
6 Taps	$x^9+x^6+x^4+x^3+x^2+x+1$, $x^9+x^6+x^5+x^3+x^2+x+1$, $x^9+x^6+x^5+x^4+x^2+x+1$, $x^9+x^6+x^5+x^4+x^3+x^2+1$ $x^9+x^7+x^5+x^3+x^2+x+1$, $x^9+x^7+x^5+x^4+x^2+x+1$, $x^9+x^7+x^5+x^4+x^3+x^2+1$, $x^9+x^7+x^6+x^3+x^2+x+1$, $x^9+x^7+x^6+x^4+x^3+x+1$, $x^9+x^7+x^6+x^5+x^4+x^2+1$, $x^9+x^7+x^6+x^5+x^4+x^3+1$, $x^9+x^8+x^4+x^3+x^2+x+1$, $x^9+x^8+x^5+x^4+x^3+x+1$, $x^9+x^8+x^6+x^3+x^2+x+1$, $x^9+x^8+x^6+x^4+x^3+x+1$, $x^9+x^8+x^6+x^5+x^3+x+1$, $x^9+x^8+x^6+x^5+x^3+x^2+1$, $x^9+x^8+x^6+x^5+x^4+x+1$, $x^9+x^8+x^7+x^3+x^2+x+1$, $x^9+x^8+x^7+x^5+x^4+x^2+1$, $x^9+x^8+x^7+x^5+x^4+x^3+1$, $x^9+x^8+x^7+x^6+x^2+x+1$, $x^9+x^8+x^7+x^6+x^3+x+1$, $x^9+x^8+x^7+x^6+x^3+x^2+1$, $x^9+x^8+x^7+x^6+x^4+x^2+1$, $x^9+x^8+x^7+x^6+x^4+x^3+1$, $x^9+x^8+x^7+x^6+x^5+x+1$, $x^9+x^8+x^7+x^6+x^5+x^3+1$,
8 Taps	$x^9+x^8+x^6+x^5+x^4+x^3+x^2+x+1$ $x^9+x^8+x^7+x^6+x^5+x^4+x^3+x+1$

Running the above algorithm a set of minimum to maximum shift register feedback taps based PN sequence generator of size n = 2 to 10 are obtained. Corresponding to shift register feedback taps the primitive characteristic polynomials are listed below in Tables 2 to 6. For higher values of n > 10 it is difficult to present. Through our simulation experiment we computed all possible primitive polynomial up to n = 44. For the values n = 11 to 42 the number of taps corresponding to primitive polynomials are given in Table 7.

TABLE 6
PRIMITIVE CHARACTERISTIC POLYNOMIALS (N = 10)

n = 10	
2 Taps	$x^{10}+x^3+1$, $x^{10}+x^7+1$
4 Taps	$x^{10}+x^4+x^3+x+1$, $x^{10}+x^5+x^2+x+1$, $x^{10}+x^5+x^3+x^2+1$, $x^{10}+x^6+x^5+x^2+1$, $x^{10}+x^7+x^3+x+1$, $x^{10}+x^7+x^6+x^2+1$, $x^{10}+x^8+x^3+x^2+1$, $x^{10}+x^8+x^4+x^3+1$, $x^{10}+x^8+x^5+x+1$, $x^{10}+x^8+x^5+x^4+1$, $x^{10}+x^8+x^6+x+1$, $x^{10}+x^8+x^7+x^2+1$, $x^{10}+x^8+x^7+x^5+1$, $x^{10}+x^9+x^4+x+1$, $x^{10}+x^9+x^4+x^2+1$, $x^{10}+x^9+x^5+x^2+1$, $x^{10}+x^9+x^6+x+1$, $x^{10}+x^9+x^7+x^3+1$' $x^{10}+x^9+x^7+x^6+1$, $x^{10}+x^9+x^8+x^5+1$,
6 Taps	$x^{10}+x^6+x^5+x^3+x^2+x+1$, $x^{10}+x^7+x^6+x^4+x^2+x+1$, $x^{10}+x^7+x^6+x^5+x^4+x+1$, $x^{10}+x^8+x^5+x^4+x^3+x^2+1$, $x^{10}+x^8+x^6+x^4+x^2+x+1$, $x^{10}+x^8+x^6+x^5+x^3+x+1$, $x^{10}+x^8+x^7+x^3+x^2+x+1$, $x^{10}+x^8+x^7+x^4+x^2+x+1$, $x^{10}+x^8+x^7+x^6+x^2+x+1$, $x^{10}+x^8+x^7+x^6+x^5+x^2+1$, $x^{10}+x^9+x^5+x^4+x^2+x+1$, $x^{10}+x^9+x^6+x^3+x^2+x+1$, $x^{10}+x^9+x^6+x^4+x^3+x+1$, $x^{10}+x^9+x^6+x^5+x^4+x^3+1$, $x^{10}+x^9+x^7+x^5+x^4+x^2+1$, $x^{10}+x^9+x^7+x^6+x^4+x+1$, $x^{10}+x^9+x^8+x^4+x^2+x+1$, $x^{10}+x^9+x^8+x^4+x^3+x^2+1$, $x^{10}+x^9+x^8+x^5+x^4+x^3+1$, $x^{10}+x^9+x^8+x^6+x^2+x+1$, $x^{10}+x^9+x^8+x^6+x^3+x^2+1$, $x^{10}+x^9+x^8+x^6+x^4+x^2+1$, $x^{10}+x^9+x^8+x^6+x^4+x^3+1$, $x^{10}+x^9+x^8+x^6+x^5+x+1$, $x^{10}+x^9+x^8+x^7+x^3+x^2+1$, $x^{10}+x^9+x^8+x^7+x^4+x+1$, $x^{10}+x^9+x^8+x^7+x^5+x^4+1$
8 Taps	$x^{10}+x^7+x^6+x^5+x^4+x^3+x^2+x+1$, $x^{10}+x^8+x^7+x^6+x^5+x^4+x^2+x+1$, $x^{10}+x^8+x^7+x^6+x^5+x^4+x^3+x+1$, $x^{10}+x^9+x^6+x^5+x^4+x^3+x^2+x+1$, $x^{10}+x^9+x^7+x^6+x^4+x^3+x^2+x+1$, $x^{10}+x^9+x^7+x^6+x^5+x^4+x^3+x^2+1$, $x^{10}+x^9+x^8+x^6+x^5+x^4+x^3+x+1$, $x^{10}+x^9+x^8+x^7+x^6+x^4+x^3+x+1$, $x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x+1$, $x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+1$

TABLE 7
NUMBER OF TAPS CRRESPONDING TO PRIMITIVE CHARACTERISTIC POLYNOMIALS (N = 11 TO 42)

Taps	n
2	11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 39, 41, 42
4	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
6	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
8	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
10	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
12	13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
14	15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
16	17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
18	19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
20	21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
22	23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
24	25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
26	27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
28	29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
30	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42
32	33, 34, 35, 36, 37, 38, 39, 40, 41, 42
34	35, 36, 37, 38, 39, 40, 41, 42
36	37, 38, 39, 40, 41, 42
38	39, 40, 41, 42
40	41, 42

IV. VERILOG – HDL LFSR SAMPLE MODEL

As a sample example a 5-bit PN sequence generation is considered to demonstrate the Verilog – HDL implementation.

```
module LFSR (CLOCK, RESET, DIN, LOAD, SHIFT, Q);
parameter N = n;
parameter tap_0 = 0;
parameter tap_i = i;
input   CLK;
input   RESET;
input   DIN;
input   LOAD;
input   SHIFT;
output  [N-1:0]  Q;
reg     [N-1:0]  Q;
reg     [N-1:0]  Q_N;
wire    TAPS;
always @ (posedge CLK or posedge RESET)
begin
if (RESET)
Q <= #1 0-1;
else
Q <= #1 Q_N;
end
assign TAPS = Q[tap_0] ^ . . .^ Q[tap_i];
always @ (Q or SHIFT or LOAD or DIN)
begin
Q_N = Q;
if (LOAD)
Q_N = {DIN, Q[N-1:1]};
else
if (SHIFT)
Q_N = {TAPS, Q[N-1:1]};
end
endmodule
```

LFSR shifts data on rising (positive) edge of clock whenever SHIFT or LOAD (load control) is high. Q is the output of the LFSR. When LOAD is high, Serial Load Data IN (DIN) is shifted into the register. Data is shifted from MSB to 0 of Q. RESET represents the asynchronous reset. Running the verilog code for n = 5, seed load of [11111]: the results in the shifts of the states are given below as RUN 1 and RUN 2 for the TAPS = [T_0, T_3, T_5], and TAPS = [T_0, T_2, T_3, T_4, T_5] respectively.

RUN 1:

5'h1F →5'h1E →5'h1C →5'h19 →5'h12 → 5'h04 → 5'h09 → 5'h13 → 5'h06 → 5'h0C → 5'h18 →5'h10 → 5'h01 → 5'h02 → 5'h05 → 5'h0B → 5'h16 → 5'h0D → 5'h1A → 5'h15 → 5'h0A → 5'h14 → 5'h08 → 5'h11 → 5'h03 → 5'h07 → 5'h0E →5'h1D →5'h1B → 5'h17 → 5'h0F

RUN 2:

5'h1F →5'h1E → 5'h1C →5'h18 →5'h11 →5'h03 → 5'h06 → 5'h0D → 5'h1B → 5'h17 → 5'h0E → 5'h1D

→5'h1A → 5'h15 → 5'h0A → 5'h14 → 5'h08 → 5'h10 → 5'h01 → 5'h02 → 5'h04 → 5'h09 → 5'h12 → 5'h05 → 5'h0B → 5'h16 → 5'h0C → 5'h19 →5'h13 →5'h07 → 5'h17 → 5'h0F

V. ANALYSIS OF LFSR DESIGN

Different structures of LFSRs for n = 2 – 7, 9 – 15, 17 – 23, 25 – 31, 33 – 39 and 41 – 42 can generate m-sequences with the number of taps ranging from 2 to 40. Whereas for n = 8, 16, 24, 32, 40 can be realized with the number of taps ranging from 4 to 40. A comparative study is carried out to judge the impact of hardware with the minimum and maximum number of taps. Fig. 2 demonstrates such an example for values n = 2 to 20. Fig. 3 shows extra hardware requirements whereas Fig. 4 demonstrates extra hardware requirement per unit length of PN generator.

VI. CONCLUSION

As it can be viewed from the simulation results of our study shown in Figures 2 - 4 that proper tap selection reduces hardware complexity to a marginal level. We also want to highlight that the selection of higher lengths of LFSRs with sparse feedback coefficients gives much better attributes in comparison with the selections of lower lengths of the LFSRs. For example, in Fig. 4, it can be visualized that the selection of an 8-bit LFSR is much more economical than choosing 5 or 6 or 7 bit LFSR. Similar case occurs in 12-bit LFSR which is better than 9 or 10 or 11-bit size. Whereas 19 is a better choice amongst 15 to 19 bit sizes， as it can be seen in Figure 4.

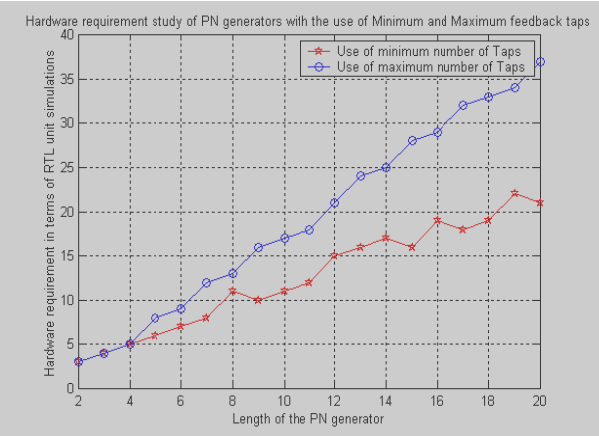

Fig. 2 Hardware requirement

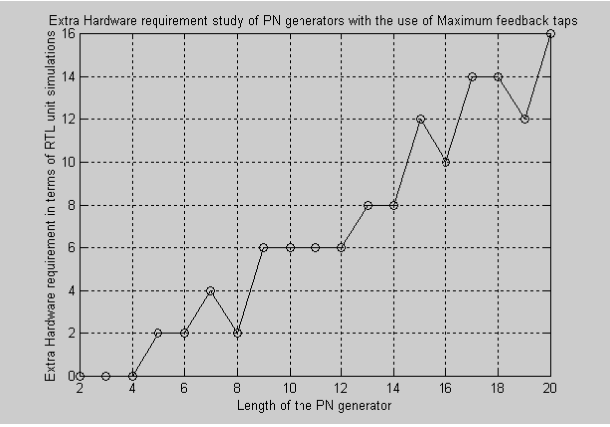

Fig. 3 Hardware requirement ratio

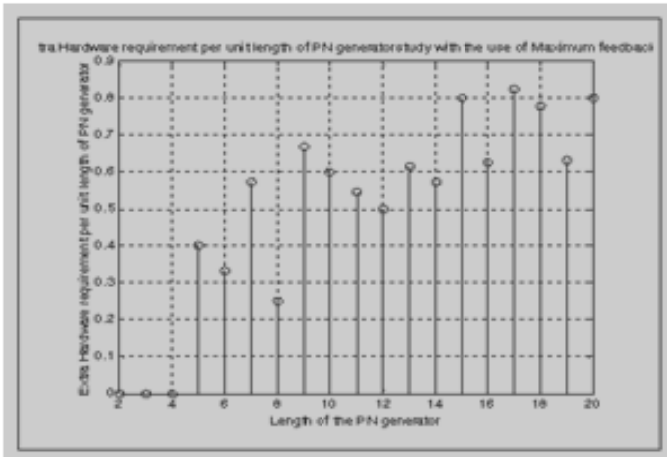

Fig. 4 Hardware requirement per unit of PN size

ACKNOWLEDGMENT

Thanks are due and acknowledged to Sultan Qaboos University (Sultanate of Oman) for providing research support grant (SQU-DVC/ PSR/RAID/2010/2).

REFERENCES

[1] T. S. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall, NJ, 1996.

[2] W. Stallings, Data and Computer Communications, Prentice Hall, Englewood, NJ, 1999.

[3] Duel-Hallen, J. Holtzman,and Z. Zvonar, Multiuser Detection for CDMA Systems, IEEE Personal Communications, pp. 46–58, April 1995.

[4] S. Moshavi, Multi-User Detection for DS-CDMA Communications, IEEE Communications Magazine, pp. 124–136, October 1996.

[5] S. Verd´u, Multiuser Detection, Cambridge University Press, Cambridge, UK, 1998.

[6] E. H. Dinan and B. Jabbari, Spreading Codes for Direct Sequence CDMA and Wideband CDMA Cellular Networks, IEEE Communications Magazine, pp. 48–54,September 1998.

[7] J. H. Lindholm, An Analysis of the Pseudo Randomness Properties of Subsequences of Long m-Sequences, IEEE Transactions on Information Theory, vol. IT-14, pp. 569–576, July 1968.

[8] D. V. Sarwate and M. B. Pursley, Crosscorrelation Properties of Pseudorandom and Related Sequences, Proceedings of the IEEE, vol. 68, no. 5,pp. 593–619, May 1980.

[9] V. Tarokh,N. Seshadri,and A. R. Calderbank, Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction, IEEE Transactions on Information Theory, vol. 44, no. 2,pp. 744 - 765, Marc h 1998.

[10] S. W. Golomb, Shift register sequences, Aegean Park Press, Leguna Hills, U.S.A. 1982.

[11] W. W. Peterson and J. J. Weldon, Error correcting codes, 2nd edition, MIT Press, Cambridge, London 1972.

[12] Ahmad, A., A Simulation Experiment on a Built-In Self Test Equipped with Pseudorandom Test Pattern Generator and Multi-Input Shift Register (MISR), International Journal of VLSI Design & Communication Systems, vol. 1, no. 4, pp. 1-12, 2010.

[13] Ahmad, A., Dawood Al-Abri, Design of an Optimal Test Simulator for Built-In Self Test Environment, Journal of Engineering Research, vol. 7, no. 2, pp. 69 – 79, 2010.

[14] Ahmad A., Testing of complex integrated circuits (ICs) – The bottlenecks and solutions, Asian Journal of Information Technology, vol. 4, no. 9, pp. 816 – 822, 2005.

[15] Ahmad, A. and Al-Habsi, A. H., Design of a built-in multi-mode ICs tester with higher testability features- A most suitable testing tool for BIST environment, Journal of IETE Technical Review, vol. 15, no. 3, pp. 283 – 288, 1998.

[16] Nanda N.K., Ahmad A. and Gaindhar V.C., Shift register modification for multipurpose use in combinational circuit testing, International Journal of Electronics (UK), vol.66, no.6, pp. 875 – 878, 1989.

[17] Ahmad A. and Nanda N.K., Effectiveness of multiple compressions of multiple signatures, International Journal of Electronics (UK), vol.66, no.5, pp.775 – 787, 1989.

[18] Ahmad A., Achievement of higher testability goals through the modification of shift register in LFSR based testing, International Journal of Electronics (UK), vol. 82, no. 3, pp. 249-260, 1997.

[19] Ahmad A., Al-Lawati A. M. J., Jervase J. A. and Zabalawi, I. H., The study of the effect of rotationally delayed transmission of data on error masking behavior of different types of signature analysis schemes, Journal for Scientific Research - Science and Technology, vol. 1, p. 88, 1996.

[20] Ali Al-Lawati and Ahmad, A., Realization of a simplified controllability computation procedure – A MATLAB-SIMULINK based tool, Journal for Scientific Research - Science and Technology, Oman, vol. 8, 2004, pp. 131 – 143**,** 2004.

[21] Ahmad A, Al-Lawati, A. M. J. and Ahmed M. Al-Naamany, Identification of test point insertion location via comprehensive knowledge of digital system's nodal controllability through a simulated tool, Asian Journal of Information Technology (AJIT), vol. 3, no. 3, pp. 142 – 147, 2004.

[22] Ahmad, A., Investigation of a constant behavior aliasing errors in signature analysis due to the use of different ordered test-patterns in a BIST testing techniques, Journal of Microelectronics and Reliability, (PERGAMON, Elsevier Science), vol. 42, pp. 967 – 974, 2002.

[23] Ahmad, A., Constant error masking behavior of an internal XOR type signature analyzer due to the changed polynomial seeds, Journal Of Computers & Electrical Engineering (PERGAMON, Elsevier Science), vol. 28, no. 6, pp. 577 – 585, 2002.

[24] Al-Naamany, A. M., and Ahmad A., Development of a strong stream ciphering technique using non-linear fuzzy logic selector, Mobile and Wireless Communications, Kluwer Academic Publishers (Reference: PWC'02 – IFIP 106; Singapore), pp. 199 – 206, 2002.

[25] Ahmad A., Al-Musharafi. M. J., Al-Busaidi, S., Design and study of a strong stream crypto-system model for e-commerce, International Council for Computer Communication Publishers – Washington DC, USA (The ACM Library), pp. 619 – 630, 2002.

[26] Ahmad A., Critical role of polynomial seeds on the effectiveness of an LFSR-based testing technique, International Journal of Electronics (UK), vol.77, no.2, pp.127 – 137, 1994.

[27] Ahmad A., Nanda N.K. and Garg K., Are primitive polynomials always best in signature analysis?, IEEE design & Test of Computers (USA), vol.7, no.4, pp. 36 – 38, 1990.

[28] Ahmad A., Nanda N.K. and Garg K., A critical role of primitive polynomials in an LFSR based testing technique, IEE Electronics Letters (UK), vol.24, no.15, pp. 953 – 955, 1988.

[29] Ahmad A., Nanda N.K. and Garg K., The use of irreducible characteristic polynomials in an LFSR based testing of digital circuits, Proceedings of 4th IEEE int'l conference (TENCON-89), held at Bombay (India), Nov. 21-23, pp. 494-496, 1989.

[30] Ahmad, A., Investigation of Typical Properties of Some LFSR Structures, Journal of System Science and Engineering, vol. 17, no. 1, pp. 65 – 69, 2008.

[31] Ahmad, A., and Al-Maashri, A., Investigating Some Special Sequence Length Generated Through an External Exclusive-NOR Type LFSRs, International Journal Electrical and Computer Engineering, (PERGAMON, Elsevier Science), vol. 34, pp. 270 – 280, 2008.

[32] Ahmad, A., Development of State Model Theory for External Exclusive NOR Type LFSR Structures, Enformatika, Volume 10, pp. 125 – 129, 2005.

[33] T. Jamil and Ahmad A., An investigation in to the application of linear feedback shift registers for steganography, Proceedings IEE SoutheastCon2002, pp. 239 – 244, 2002.

[34] Ahmad, A., Al-Musharafi, M.J., and Al-Busaidi S., A new algorithmic procedure to test m-sequences generating feedback connections of stream cipher's LFSRs, - IEEE 01CH37239 (TENCON'01), pp. 366 – 369, 201.

[35] Ahmad A. and Elabdalla A. M., An efficient method to determine linear feedback connections in shift registers that generate maximal length pseudo-random up and down binary sequences, Computer & Electrical Engineering -An Int'l Journal (USA), vol. 23, no. 1, pp. 33-39, 1997.